Linear regression Advertising data

```
# Packages
library(tidyverse)  # data manipulation and visualization

## Warning: package 'tidyverse' was built under R version 3.4.3

## -- Attaching packages ---------------------------------------------------
----------------------------------------------------------- tidyverse 1.2.1 -
-

## v ggplot2 3.1.0     v purrr   0.2.4
## v tibble  2.0.1     v dplyr   0.7.4
## v tidyr   0.8.0     v stringr 1.2.0
## v readr   1.1.1     v forcats 0.3.0

## Warning: package 'ggplot2' was built under R version 3.4.4

## Warning: package 'tibble' was built under R version 3.4.4

## Warning: package 'tidyr' was built under R version 3.4.3

## Warning: package 'readr' was built under R version 3.4.3

## Warning: package 'purrr' was built under R version 3.4.3

## Warning: package 'dplyr' was built under R version 3.4.3

## Warning: package 'forcats' was built under R version 3.4.3

## -- Conflicts ------------------------------------------------------------
----------------------------------------------------- tidyverse_conflicts() -
-
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()

library(modelr)     # provides easy pipeline modeling functions

## Warning: package 'modelr' was built under R version 3.4.3

library(broom)      # helps to tidy up model outputs

## Warning: package 'broom' was built under R version 3.4.3

##
## Attaching package: 'broom'

## The following object is masked from 'package:modelr':
##
##     bootstrap
```

Load the Data:

```
Advertising<-read.csv("Advertising.csv")
```

Preparing our Data

-Diving the data into train and test

```
set.seed(123)
sample <- sample(c(TRUE, FALSE), nrow(Advertising), replace = T, prob =
c(0.6,0.4))
train <- Advertising[sample, ]
test <- Advertising[!sample, ]
```

Simple Linear Regression

Y = β0 + β1X + ϵ

where:

Y represents sales X represents TV advertising budget β 0 is the intercept β 1 is the coefficient (slope term) representing the linear relationship ϵ is a mean-zero random error term

Model Building To build this model in R we use the formula notation of Y ~ X

```
model1 <- lm(Sales ~ TV, data = train)

summary(model1)

##
## Call:
## lm(formula = Sales ~ TV, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.5816 -1.7845 -0.2533  2.1715  6.9345
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 6.764098   0.607592   11.13   <2e-16 ***
## TV          0.050284   0.003463   14.52   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.204 on 120 degrees of freedom
## Multiple R-squared:  0.6373, Adjusted R-squared:  0.6342
## F-statistic: 210.8 on 1 and 120 DF,  p-value: < 2.2e-16
```

Y= 6.76 + 0.05X +ϵ

```
tidy(model1)
```

```
##          term   estimate std.error statistic      p.value
## 1 (Intercept) 6.76409784 0.6075916  11.13264 3.307215e-20
## 2          TV 0.05028368 0.0034632  14.51943 3.413075e-28
```

```r
confint(model1)
```

```
##                  2.5 %     97.5 %
## (Intercept) 5.56110868 7.96708701
## TV          0.04342678 0.05714057
```

Our results show us that our 95% confidence interval for β1 (TV) is [.043, .057].

RSE:

```r
sigma(model1)
```

```
## [1] 3.204129
```

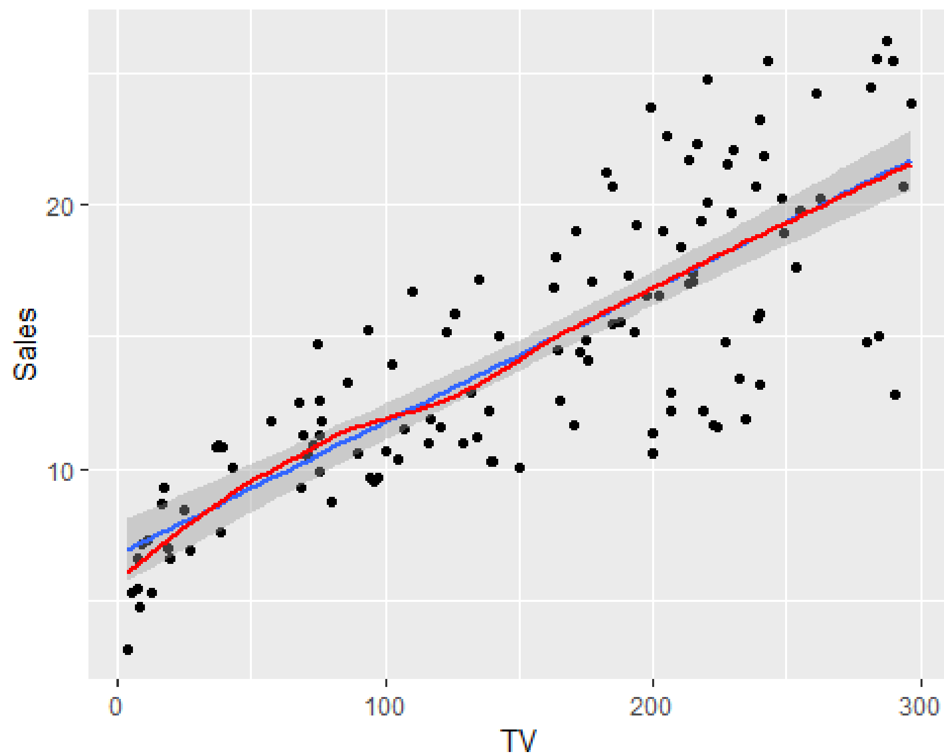
R^2 (R-Square)

```r
rsquare(model1, data = train)
```

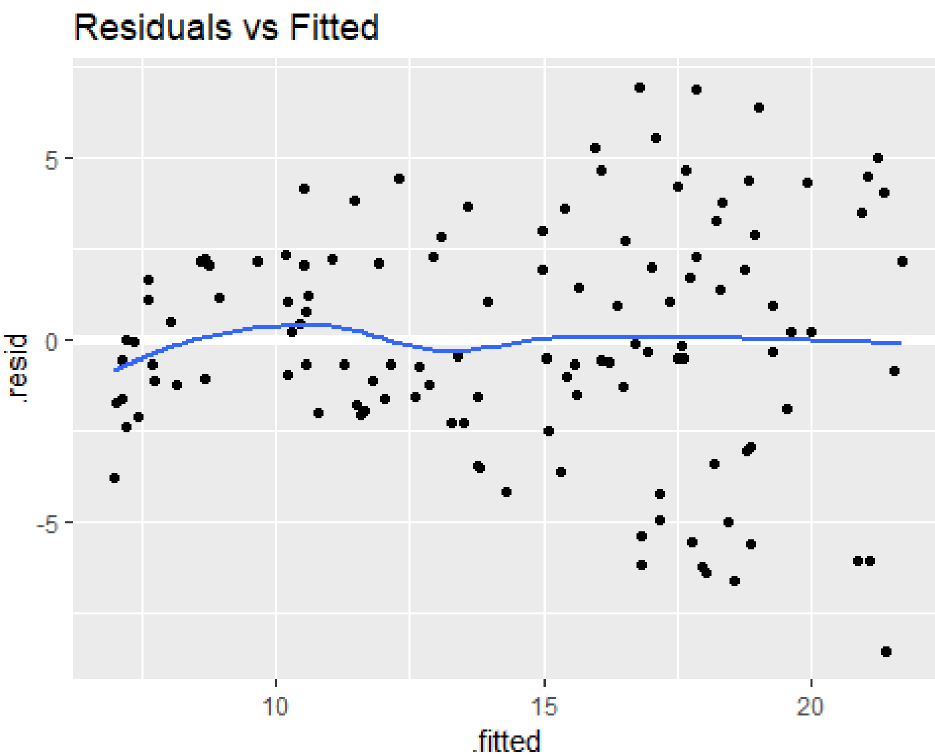
```
## [1] 0.6372581
```

```r
ggplot(train, aes(TV, Sales)) +
  geom_point() +
  geom_smooth(method = "lm") +
  geom_smooth(se = FALSE, color = "red")
```

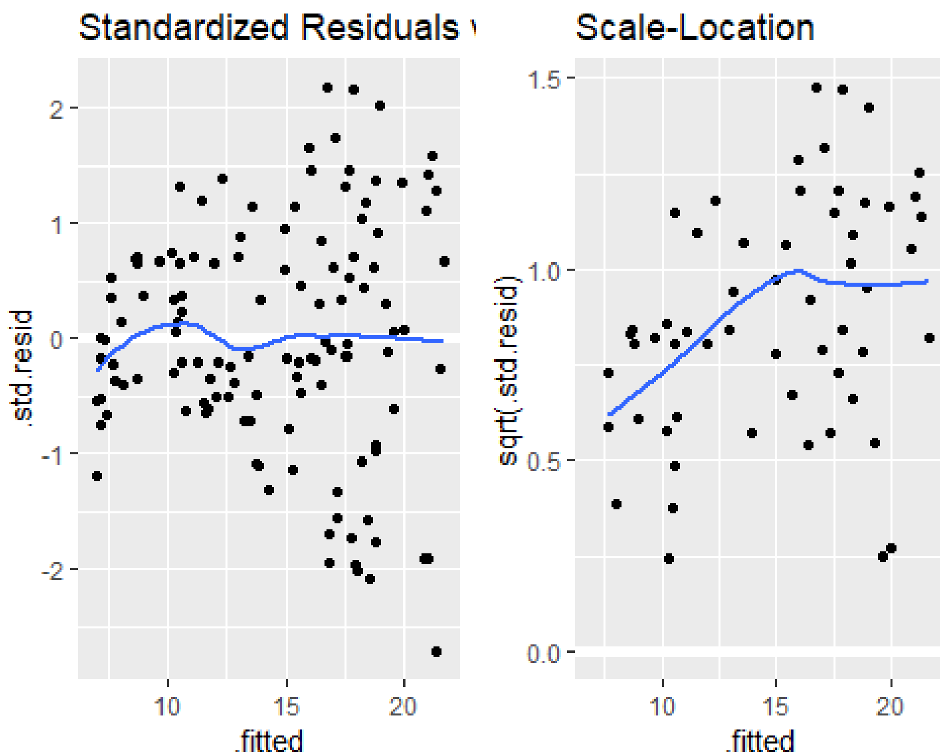
```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```




```r
# add model diagnostics to our training data
model1_results <- augment(model1, train)

ggplot(model1_results, aes(.fitted, .resid)) +
  geom_ref_line(h = 0) +
  geom_point() +
  geom_smooth(se = FALSE) +
  ggtitle("Residuals vs Fitted")

## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```




```
p1 <- ggplot(model1_results, aes(.fitted, .std.resid)) +
  geom_ref_line(h = 0) +
  geom_point() +
  geom_smooth(se = FALSE) +
  ggtitle("Standardized Residuals vs Fitted")

p2 <- ggplot(model1_results, aes(.fitted, sqrt(.std.resid))) +
  geom_ref_line(h = 0) +
  geom_point() +
  geom_smooth(se = FALSE) +
  ggtitle("Scale-Location")

gridExtra::grid.arrange(p1, p2, nrow = 1)

## `geom_smooth()` using method = 'loess' and formula 'y ~ x'

## Warning in sqrt(.std.resid): NaNs produced

## Warning in sqrt(.std.resid): NaNs produced

## `geom_smooth()` using method = 'loess' and formula 'y ~ x'

## Warning: Removed 65 rows containing non-finite values (stat_smooth).

## Warning: Removed 65 rows containing missing values (geom_point).
```



Multiple Regression

```
model2 <- lm(Sales ~ TV + Radio + Newspaper, data = train)

summary(model2)

##
## Call:
## lm(formula = Sales ~ TV + Radio + Newspaper, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.8426 -0.6466  0.2165  1.0640  2.6804
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.822206   0.369369   7.641 6.29e-12 ***
## TV           0.047362   0.001657  28.577  < 2e-16 ***
## Radio        0.196375   0.010347  18.979  < 2e-16 ***
## Newspaper   -0.010593   0.006460  -1.640    0.104
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.527 on 118 degrees of freedom
## Multiple R-squared:  0.9189, Adjusted R-squared:  0.9169
## F-statistic: 445.9 on 3 and 118 DF,  p-value: < 2.2e-16
```

Coefficients for TV and Radio advertising budget are statistically significant (p-value < 0.05) while the coefficient for Newspaper is not. Thus, changes in Newspaper budget do not appear to have a relationship with changes in sales.

```
tidy(model2)
```

```
##          term    estimate   std.error statistic      p.value
## 1 (Intercept)  2.82220600 0.369369200   7.64061 6.292503e-12
## 2          TV  0.04736209 0.001657356  28.57690 7.360519e-55
## 3       Radio  0.19637507 0.010346756  18.97939 1.171088e-37
## 4   Newspaper -0.01059255 0.006460332  -1.63963 1.037455e-01
```

```
confint(model2)
```

```
##                   2.5 %      97.5 %
## (Intercept)  2.09075443 3.553657581
## TV           0.04408008 0.050644109
## Radio        0.17588568 0.216864467
## Newspaper   -0.02338577 0.002200661
```

Assessing Model Accuracy

```
list(model1 = broom::glance(model1), model2 = broom::glance(model2))
```

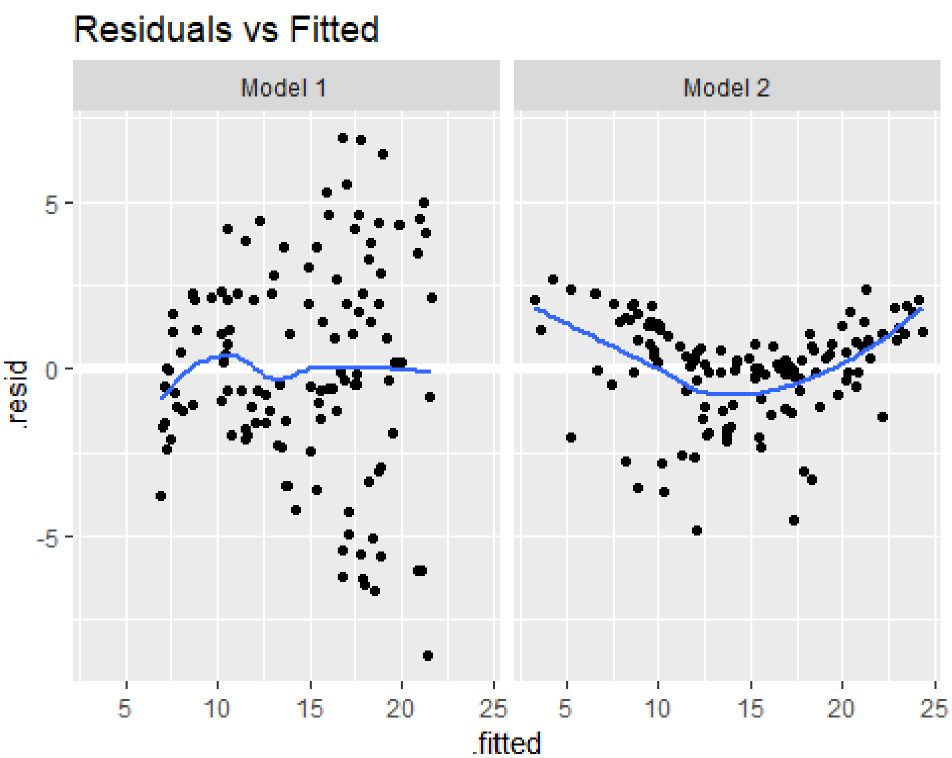
```
## $model1
##   r.squared adj.r.squared    sigma statistic      p.value df    logLik
## 1 0.6372581     0.6342353 3.204129  210.8137 3.413075e-28  2 -314.1639
##        AIC      BIC deviance df.residual
## 1 634.3279 642.7399 1231.973         120
## 
## $model2
##   r.squared adj.r.squared    sigma statistic      p.value df    logLik
## 1 0.9189394     0.9168785 1.527446  445.9001 3.486405e-64  4 -222.7558
##        AIC      BIC deviance df.residual
## 1 455.5116 469.5317 275.3046         118
```

1. R^2: Model 2's R^2 =.92 is substantially higher than model 1 suggesting that model 2 does a better job explaining the variance in sales.
2. RSE: Model 2's RSE (sigma) is lower than model 1. This shows that model 2 reduces the variance of our ϵ parameter which corroborates our conclusion that model 2 does a better job modeling sales.
3. F-statistic: the F-statistic (statistic) in model 2 is larger than model 1. Here larger is better and suggests that model 2 provides a better "goodness-of-fit".

Assessing Our Model Visually

```
# add model diagnostics to our training data
model1_results <- model1_results %>%
  mutate(Model = "Model 1")

model2_results <- augment(model2, train) %>%
```

```
  mutate(Model = "Model 2") %>%
  rbind(model1_results)

ggplot(model2_results, aes(.fitted, .resid)) +
  geom_ref_line(h = 0) +
  geom_point() +
  geom_smooth(se = FALSE) +
  facet_wrap(~ Model) +
  ggtitle("Residuals vs Fitted")

## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



Making Predictions

```
test %>%
  gather_predictions(model1, model2) %>%
  group_by(model) %>%
  summarise(MSE = mean((Sales-pred)^2))

## Warning: package 'bindrcpp' was built under R version 3.4.3

## Warning: `as_dictionary()` is soft-deprecated as of rlang 0.3.0.
## Please use `as_data_pronoun()` instead
## This warning is displayed once per session.

## Warning: `new_overscope()` is soft-deprecated as of rlang 0.2.0.
## Please use `new_data_mask()` instead
## This warning is displayed once per session.
```

```
## Warning: The `parent` argument of `new_data_mask()` is deprecated.
## The parent of the data mask is determined from either:
##
##   * The `env` argument of `eval_tidy()`
##   * Quosure environments when applicable
## This warning is displayed once per session.

## Warning: `overscope_clean()` is soft-deprecated as of rlang 0.2.0.
## This warning is displayed once per session.

## # A tibble: 2 x 2
##   model    MSE
##   <chr>  <dbl>
## 1 model1 11.3
## 2 model2  3.75
```

Is there synergy among the advertising media?

```
# option A
model3 <- lm(Sales ~ TV + Radio + TV * Radio, data = train)

# option B
model3 <- lm(Sales ~ TV * Radio, data = train)

tidy(model3)

##          term    estimate    std.error statistic      p.value
## 1 (Intercept) 6.497388545 3.078842e-01 21.103355 7.247874e-42
## 2          TV 0.020790280 1.875342e-03 11.086126 5.251042e-20
## 3       Radio 0.039032099 1.058511e-02  3.687455 3.437776e-04
## 4    TV:Radio 0.001014227 6.208425e-05 16.336294 4.468987e-32
```

Assessing Model Accuracy

```
list(model1 = broom::glance(model1),
     model2 = broom::glance(model2),
     model3 = broom::glance(model3))

## $model1
##   r.squared adj.r.squared    sigma statistic      p.value df    logLik
## 1 0.6372581     0.6342353 3.204129  210.8137 3.413075e-28  2 -314.1639
##        AIC      BIC deviance df.residual
## 1 634.3279 642.7399 1231.973         120
##
## $model2
##   r.squared adj.r.squared    sigma statistic      p.value df    logLik
## 1 0.9189394     0.9168785 1.527446  445.9001 3.486405e-64  4 -222.7558
##        AIC      BIC deviance df.residual
## 1 455.5116 469.5317 275.3046         118
##
## $model3
```

```
##   r.squared adj.r.squared     sigma statistic      p.value df    logLik
## 1 0.9745811     0.9739349 0.8553403  1508.073 6.908026e-94  4 -152.0138
##        AIC      BIC deviance df.residual
## 1 314.0275 328.0476 86.32963         118
```

We can compare our model results across all three models. We see that our adjusted R2 and F-statistic are highest with model 3 and our RSE, AIC, and BIC are the lowest with model 3; all suggesting the model 3 out performs the other models.

Assessing Our Model Visually

```
# add model diagnostics to our training data
model3_results <- augment(model3, train) %>%
  mutate(Model = "Model 3") %>%
  rbind(model2_results)

ggplot(model3_results, aes(.fitted, .resid)) +
  geom_ref_line(h = 0) +
  geom_point() +
  geom_smooth(se = FALSE) +
  facet_wrap(~ Model) +
  ggtitle("Residuals vs Fitted")

## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```